IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application. No: 10/051,474	§	
Filed: January 18, 2002	§	Atty. Dkt. No: 5150-54200
Inventor(s): Sundeep Chandhoke, Nicolas Vazquez, David W Fuller and Christopher Cifra	§	
Title: SYSTEM AND METHOD FOR GRAPHICALLY CREATING A SEQUENCE OF MOTION CONTROL OPERATIONS	§	
Examiner: Pillai, Namitha	§	
Group/Art Unit: 2173	§	

APPEAL BRIEF

Dear Sir/Madam:

Further to the Notice of Appeal filed December 8, 2006 and the Notice of Non-Compliant Appeal Brief mailed May 21, 2007, Appellant presents this amended Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-48 were submitted in the application. Claims 4, 8-10, 24, and 26 have been canceled. Claims 1-3, 5-7, 11-23, 25, and 27-48 remain pending in the application. (Appellant notes that the current Office Action identifies claim 10 as a pending claim. However, claim 10 was canceled in the amendment submitted on February 27, 2006.)

All of the pending claims (1-3, 5-7, 11-23, 25, and 27-48) stand rejected and are the subject of this appeal. A copy of the claims, as incorporating entered amendments and as on appeal, is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been filed subsequent to the amendment of October 26, 2005. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The subject matter of the present claims relates generally to the field of graphical software programming. A graphical program is typically created by including various function nodes or icons in a block diagram and interconnecting them, e.g., by drawing lines or wires between them. The resulting interconnected nodes visually indicate functionality of the graphical program, e.g., visually indicate a function or process performed by the graphical program during its execution.

More particularly, the present claims relate to automatically generating a graphical program, e.g., automatically including a plurality of nodes in the graphical program and automatically generating connections between the nodes, such that the interconnected nodes visually indicate functionality of the graphical program.

More particularly still, the present claims relate to automatically generating a graphical program in order to implement a sequence of motion control operations that has been specified by a user. For example, the method of claim 1 comprises displaying a graphical user interface that provides graphical user interface access to a set of motion control operations, e.g., operations for controlling a motion control device coupled to a computer system. (*See 401 of Figure 4; Figures 6A – 6F; p. 20, lines 12 – 25*).

The method further comprises receiving user input to the graphical user interface, where the user input specifies a sequence of motion control operations. (*See 403 of Figure 4; Figure 5; Figures 6A – 6F; p. 22, lines 5-16; p. 26, line 6 – p. 28, line 17*). For example, the user may interact with the displayed graphical user interface in order to select various motion control operations to be included in the sequence, e.g., for a particular motion control application.

The method further comprises automatically generating a graphical program implementing the specified sequence of motion control operations. (*See 405B of Figure 4; Figure 7; Figures 8A – 8G; p. 24, lines 4-23; p. 44, line 23-p. 51, line 3*). As well known in the prior art in the field of graphical programming, a user typically manually creates a graphical program by selecting various function nodes or icons and interconnecting them, e.g., by drawing lines or wires between them. However, in the method of claim 1, the graphical program is automatically generated. Automatically

generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, where the interconnected nodes visually indicate functionality of the graphical program.

The method further comprises performing the specified sequence of motion control operations. (*See 405 and 405A-405E of Figure 4; p. 22, line 24 – p. 25, line 27*).

Figures 6A – 6F may be helpful in understanding the above description of the method of claim 1. Figures 6A – 6F illustrate an exemplary graphical user interface with which a user may interact in order to specify a sequence of motion control operations. The user presses the buttons 600 (see Figure 6A) in order to add the desired motion control operations to the sequence. Each motion control operation that has been added to the sequence is represented by a respective icon 604 in the icon strip 602 (see Figure 6A). Figure 6F illustrates the sequence of motion control operations after the user has added four motion control operations, represented as four respective icons 604 in the icon strip 602. (Note that these icons are not graphical program nodes, but simply represent the motion control operations that have been added to the sequence.)

Figures 8A – 8G illustrate a graphical program that has been automatically generated based on the sequence of motion control operations specified by the user in Figures 6A – 6F. (Figures 8A – 8G illustrate a single graphical program, but the size of the program requires it to be separated into multiple drawings.) Thus, the user interacts with the graphical user interface shown in Figures 6A – 6F to create a desired sequence of operations, and the graphical program of Figures 8A – 8G is automatically generated, e.g., where the graphical program is executable to perform the sequence of operations which the user specified.

Independent claim 29 is a method claim that recites similar limitations as claim 1. The method comprises displaying a graphical user interface that provides graphical user interface access to a set of motion control operations. (*See 401 of Figure 4; Figures 6A – 6F; p. 20, lines 12 – 25*).

The method further comprises receiving user input to the graphical user interface, where the user input specifies a sequence of motion control operations. (*See 403 of Figure 4; Figure 5; Figures 6A – 6F; p. 22, lines 5-16; p. 26, line 6 – p. 28, line 17*).

The method further comprises automatically generating a graphical program implementing the specified sequence of motion control operations. (*See 405B of Figure 4; Figure 7; Figures 8A – 8G; p. 24, lines 4-23; p. 44, line 23-p. 51, line 3*).

The method further comprises storing information representing the specified sequence of motion control operations in a data structure. (*See p. 22, lines 17-23*).

Independent claim 32 is a method claim that recites similar limitations as claim 1. The method comprises receiving user input specifying a desired sequence of motion control operations. (*See 403 of Figure 4; Figure 5; Figures 6A – 6F; p. 22, lines 5-16; p. 26, line 6 – p. 28, line 17*).

The method further comprises recording the specified sequence of motion control operations in a data structure. The specified sequence of motion control operations comprises a motion control prototype. The motion control prototype is useable to control a motion device. (*See p. 22, lines 17-23; p. 13, line 15 – p. 14, line 27*).

The method further comprises automatically generating a graphical program implementing the specified sequence of motion control operations. (*See 405B of Figure 4; Figure 7; Figures 8A – 8G; p. 24, lines 4-23; p. 44, line 23-p. 51, line 3*).

Independent claim 35 is directed to a memory medium that stores program instructions. The program instructions are executable to display a graphical user interface that provides graphical user interface access to a set of motion control operations, e.g., operations for controlling a motion control device coupled to a computer system. (*See 401 of Figure 4; Figures 6A – 6F; p. 20, lines 12 – 25*).

The program instructions are further executable to receive user input to the graphical user interface, where the user input specifies a sequence of motion control operations. (*See 403 of Figure 4; Figure 5; Figures 6A – 6F; p. 22, lines 5-16; p. 26, line 6 – p. 28, line 17*). For example, the user may interact with the displayed graphical user

interface in order to select various motion control operations to be included in the sequence, e.g., for a particular motion control application.

The program instructions are further executable to automatically generate a graphical program implementing the specified sequence of motion control operations. (*See 405B of Figure 4; Figure 7; Figures 8A – 8G; p. 24, lines 4-23; p. 44, line 23-p. 51, line 3*). Automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, where the interconnected nodes visually indicate functionality of the graphical program.

The program instructions are further executable to perform the specified sequence of motion control operations. (*See 405 and 405A-405E of Figure 4; p. 22, line 24 – p. 25, line 27*).

Independent claim 42 is directed to a system for creating a motion control sequence. The system comprises a processor (*See, for example, CPU 160 of Figure 3*), a memory storing program instructions (*See, for example, main memory 166 of Figure 3*), and a display device (*See, for example, computer system 82 of Figure 1 and video card 180 of Figure 3*).

The processor is operable to execute the program instructions stored in the memory to display a graphical user interface that provides graphical user interface access to a set of motion control operations on the display device. (*See 401 of Figure 4; Figures 6A – 6F; p. 20, lines 12 – 25*).

The processor is further operable to execute the program instructions stored in the memory to receive user input to the graphical user interface specifying a sequence of motion control operations. (*See 403 of Figure 4; Figure 5; Figures 6A – 6F; p. 22, lines 5-16; p. 26, line 6 – p. 28, line 17*).

The processor is further operable to execute the program instructions stored in the memory to automatically generate a graphical program implementing the specified sequence of motion control operations. (*See 405B of Figure 4; Figure 7; Figures 8A – 8G; p. 24, lines 4-23; p. 44, line 23-p. 51, line 3*). Automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program

and automatically generating a plurality of connections between the nodes, where the interconnected nodes visually indicate functionality of the graphical program.

The processor is further operable to execute the program instructions stored in the memory to execute the specified sequence of motion control operations. (*See 405 and 405A-405E of Figure 4; p. 22, line 24 – p. 25, line 27*).

Independent claim 44 is directed to a system for creating a motion control sequence. The system comprises means for displaying a graphical user interface that provides graphical user interface access to a set of motion control operations. (*See Figure 3; 401 of Figure 4; Figures 6A – 6F; p. 20, lines 12 – 25*).

The system further comprises means for receiving user input to the graphical user interface specifying a sequence of motion control operations. (*See Figure 3; 403 of Figure 4; Figure 5; Figures 6A – 6F; p. 22, lines 5-16; p. 26, line 6 – p. 28, line 17*).

The system further comprises means for automatically generating a graphical program implementing the specified sequence of motion control operations. (*See Figure 3; 405B of Figure 4; Figure 7; Figures 8A – 8G; p. 24, lines 4-23; p. 44, line 23-p. 51, line 3*). Automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, where the interconnected nodes visually indicate functionality of the graphical program.

The system further comprises means for performing the specified sequence of motion control operations. (*See Figure 3; Figures 2A and 2B; 405 and 405A-405E of Figure 4; p. 22, line 24 – p. 25, line 27*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-3, 5-7, 11-23, 25, and 27-48 stand rejected under 35 U.S.C. 103(a) as being unpatentable over "Compumotor, Motion Builder Start-Up Guide & Tutorial" (hereinafter "Compumotor") and EP Publication No. 0510514 A1 to Oka et al. (hereinafter "Oka").

VII. ARGUMENT

Claims 1-3, 5-7, 11-23, 25, and 27-48 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Compumotor and Oka. Appellant respectfully traverses these rejections for the following reasons. Different claims are addressed under respective subheadings.

Independent Claims 1, 29, 32, 35, 42, and 44

Claim 1 recites in pertinent part, "automatically generating a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, wherein the interconnected nodes visually indicate functionality of the graphical program".

In the Office Action of January 12, 2006, the Examiner asserts that, "Oka discloses a similar invention including the automatic generation of a graphical program with an objective of alleviating user interaction for creating the graphical program." Appellant respectfully disagrees and submits that the Examiner has mischaracterized Oka. Oka relates to analyzing a series of processing programs and automatically drawing a flow chart which graphically represents the processing outline (Col. 1, lines 1-14; Col. 2, lines 3-12). A flow chart is not at all the same as a graphical program. A graphical program is an executable program, whereas a flow chart is not an executable program at all, but is merely a human-readable diagram which describes a process, e.g., describes a process performed by an executable program. Oka does not teach automatically including a plurality of nodes in a graphical program or automatically generating a plurality of connections between the nodes in the graphical program, as recited in claim 1.

Appellant also respectfully reminds the Board that, "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." In re Oetiker, 977 F.2d 1443, 1446, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992). See also In re Deminski, 796 F.2d

436, 230 USPQ 313 (Fed. Cir. 1986); In re Clay, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61 (Fed. Cir. 1992). However, Oka is not in the field of Appellant's endeavor, and the problem of automatically generating a flow chart is not reasonably pertinent to the particular problem with which the inventors were concerned.

Appellant also respectfully disagrees with the Examiner's assertion on p. 2 of the Office Action of January 12, 2006 that, "Compumotor discloses automatically generating a graphical program implementing the specified sequence of motion control operations." Compumotor does not disclose automatically generating a graphical program as asserted by the Examiner, but rather teaches the user manually creating a graphical program. As described on p. 12 of the specification of the present application,

> In the present application, the term "graphical program" or "block diagram" is intended to include a program comprising graphical code, e.g., two or more interconnected nodes or icons, wherein the interconnected nodes or icons may visually indicate the functionality of the program. The nodes may be connected in one or more of a data flow, control flow, and/or execution flow format. The nodes may also be connected in a "signal flow" format, which is a subset of data flow. Thus the terms "graphical program" or "block diagram" are each intended to include a program comprising a plurality of interconnected nodes or icons which visually indicate the functionality of the program. (*Emphasis added*)

Compumotor does not teach or suggest automatically generating a graphical program. In Compumotor, the user manually creates a graphical program. The user manually selects various icons or nodes and includes and positions them in the graphical program (see p. 6 and pp. 70-87). Thus, the icons or nodes in the graphical program are not automatically included in the graphical program, but instead are included in the graphical program in response to user input requesting their inclusion. In contrast, claim 1 recites that automatically generating the graphical program comprises "automatically including a plurality of nodes in the graphical program".

Also, in Compumotor, the user manually creates connections between the icons or nodes. For example, p. 87 illustrates an example where a user creates a connection between two icons by using a mouse device to click on the output side of one icon and the input side of the other icon. Thus, the connections between icons in the graphical program are not automatically generated, but instead are generated in response to user

input specifying the connections. In contrast, claim 1 recites that automatically generating the graphical program comprises "automatically generating a plurality of connections between the nodes".

Thus, Appellant respectfully submits that Compumotor does not teach automatically generating a graphical program, i.e., a program comprising a plurality of interconnected nodes which visually indicate the functionality of the program, as asserted by the Examiner. Instead, the user manually creates the graphical program.

Oka does not remedy the deficient teachings of Compumotor to make the subject matter of automatically generating a graphical program obvious, since Oka does not even relate to the field of graphical programming or teach the concept of a graphical program, but rather, relates to analyzing a series of processing programs and automatically drawing a flow chart, as described above. Thus, the cited references, taken either singly or in combination, do not teach automatically generating a graphical program.

Furthermore, claim 1 recites the following limitations: "receiving user input to the graphical user interface specifying a sequence of motion control operations" and "automatically generating a graphical program implementing the specified sequence of motion control operations". The sequence of motion control operations specified by the user, and the graphical program that is automatically generated in order to implement the sequence of motion control operations, are two different things. For example, Figures 6A – 6F illustrate an exemplary graphical user interface with which the user may interact in order to specify a sequence of motion control operations. The user presses the buttons 600 (see Figure 6A) in order to add the desired motion control operations to the sequence. Each motion control operation that has been added to the sequence is represented by a respective icon 604 in the icon strip 602 (see Figure 6A). Figure 6F illustrates the sequence of motion control operations after the user has added four motion control operations, represented as four respective icons 604 in the icon strip 602. (Note that these icons are not graphical program nodes, but simply represent the motion control operations that have been added to the sequence.)

Figures 8A – 8G illustrate a graphical program that has been automatically generated based on the sequence of motion control operations specified by the user in Figures 6A – 6F. (Figures 8A – 8G illustrate a single graphical program, but the size of

the program requires it to be separated into multiple drawings.) Thus, the user interacts with the graphical user interface shown in Figures 6A – 6F to create a desired sequence of motion control operations, and the graphical program of Figures 8A – 8G is automatically generated, where the graphical program is executable to perform the sequence of motion control operations which the user specified.

Thus, the sequence of motion control operations specified by the user, and the graphical program that is automatically generated in order to implement the sequence of motion control operations, are two different things. However, the Examiner has not taken this into account and has apparently attempted to equate both the sequence of motion control operations and the graphical program to the same thing, i.e., a graphical program as taught in Compumotor. Appellant respectfully submits that Compumotor does not teach the subject matter of receiving user input to specify a sequence of motion control operations and automatically generating a graphical program implementing the specified sequence of motion control operations, as recited in claim 1.

Thus, for at least the reasons given above, Appellant respectfully submits that the cited references do not teach the subject matter recited in claim 1, and thus, claim 1 and its dependent claims are patentably distinct over the cited art. Inasmuch as independent claims 29, 32, 35, 42, and 44 recite similar limitations as discussed above with respect to claim 1, Appellant also submits that the other independent claims, and those claims respectively dependent thereon, are also patentably distinct over the cited art.

Claim 16

Claim 16 recites the following additional limitations not taught by the cited references:

> 16. (Original) The method of claim 1,
> wherein the graphical user interface includes an area which visually represents the motion control operations in the sequence;
> wherein the method further comprises:
> for each motion control operation added to the sequence, updating the area visually representing the motion control operations in the sequence to illustrate the added motion control operation.

As discussed above, Compumotor teaches that the user selects and arranges various icons in order to form a graphical program. The graphical program is displayed on the display screen and is visually updated as the user adds each icon to the graphical program. However, claim 16 recites that, "the graphical user interface includes an area which visually represents the motion control operations in the sequence". As discussed above with reference to claim 1, the sequence of motion control operations is not the same as the graphical program that is automatically generated based on the sequence of motion control operations. For example, the graphical user interface of Figures 6A-6F includes an area where the motion control operations in the sequence are visually represented as respective icons 604 in the icon strip 602. The motion control operations in the sequence are displayed separately from the graphical program of Figures 8A-8G that is automatically generated based on the sequence.

Thus, claim 16 does not refer to the display of the graphical program, but rather refers to display of the motion control operations in the sequence. Compumotor does not teach this feature, since Compumotor does not teach the subject matter of both a sequence of motion control operations and a graphical program that is generated based on the sequence, as discussed above with reference to claim 1.

Claim 19

Claim 19 recites the following additional limitations not taught by the cited references:

> 19. (Original) The method of claim 1,
> wherein the set of motion control operations includes:
> a straight line move operation;
> an arc move operation; and
> a contoured move operation.

The Examiner asserts that these limitations are taught by Compumotor. However, Appellant can find no teaching that Compumotor provides three separate motion control operations for performing straight line moves, arc moves, and contoured moves, respectively.

Claim 20

Claim 20 recites the following additional limitations not taught by the cited references:

> 20. (Original) The method of claim 1, further comprising:
> displaying one or more views of the sequence of motion control operations on the graphical user interface, wherein the one or more views graphically preview the cumulative movement specified by the sequence of motion control operations.

The Examiner asserts that these limitations are taught by Compumotor, referring to the top figure of p. 87. However, the top figure of p. 87 merely displays the various program icons that cause the movement to be performed, i.e., displays the graphical program itself. This is not at all the same as displaying a graphical preview of the actual movement. See, for example, the views 608 and 610 of Figure 6A, which illustrate exemplary two-dimensional and three-dimensional previews, respectively, of the cumulative movement specified by the sequence of motion control operations.

Claim 21

Claim 21 recites the following additional limitations not taught by the cited references:

> 21. (Previously presented) The method of claim 20,
> wherein the one or more views includes a two-dimensional position view for viewing a two-dimensional display of position data of the sequence in one or more of an XY, YZ, or ZX plane.

The Examiner asserts that these limitations are taught by Compumotor, referring to the bottom figure of p. 80. However, the bottom figure of p. 80 displays a move profile for a single move icon in the graphical program. In contrast, claim 20 (on which claim 21 depends) recites that, "the one or more views graphically preview the cumulative movement specified by the sequence of motion control operations." Compumotor does not teach graphically previewing the cumulative movement specified by the move icons in the graphical program.

Claim 27

Claim 21 recites the following additional limitations not taught by the cited references:

> 27. (Original) The method of claim 1, further comprising:
> receiving a request from a computer program to execute the sequence of motion control operations, wherein the computer program was not used to create the sequence of motion control operations; and
> executing the specified sequence of motion control operations in response to the request.

The Examiner asserts that these limitations are taught by Compumotor, referring to pp. 90-91, where Compumotor describes downloading a program to a controller. As described, the user first downloads the program to the controller and then selects "Run Main Program" from the "Run" menu, i.e., the "Run" menu of the computer program (application) that was used to create the program. In response to the user selecting the "Run Main Program" option, the computer program that was used to create the program requests the controller to execute the program. Thus, the request to execute the program is received from the computer program that was used to create the program. In contrast, claim 27 recites that the request to execute the sequence of motion control operations is received from a computer program that was <u>not</u> used to create the sequence of motion control operations. Appellant thus respectfully submits that Compumotor does not teach the limitations recited in claim 27.

<u>Claim 14</u>

Claim 14 recites the following additional limitations not taught by the cited references:

> 14. (Original) The method of claim 13, further comprising:
> automatically displaying the graphical panel in response to adding the motion control operation to the sequence.

As the Board is certainly aware, "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art

reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)" as stated in the MPEP §2142 (*emphasis added*).

As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

The Examiner has not cited any teaching or suggestion found in the prior art for modifying Compumotor to perform the feature recited in claim 14. In fact, the Examiner has not even cited any prior art that teaches the feature recited in claim 14. Thus, Appellant respectfully submits that the Examiner has not established a case of *prima facie* obviousness.

Claim 22

Claim 22 recites the following additional limitations not taught by the cited references:

> 22. (Original) The method of claim 20,
> wherein the one or more views includes a three-dimensional position view for viewing a three-dimensional display of position data of the sequence.

The criteria necessary for establishing a case of *prima facie* obviousness are discussed above with reference to claim 14. As noted, "the prior art reference (or references when combined) must teach or suggest all the claim limitations". The

Examiner has not cited any prior art that teaches the limitations recited in claim 22. Thus, Appellant respectfully submits that the Examiner has not established a case of *prima facie* obviousness.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-3, 5-7, 11-23, 25, and 27-48 was erroneous, and reversal of the Examiner's decision is respectfully requested.

The Commissioner is authorized to charge any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5150-54200/JCH.

Respectfully submitted,

/Jeffrey C. Hood/
Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: June 21, 2007 JCH/JLB

IX. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments and as on appeal.

1. (Previously Presented) A computer-implemented method for creating a motion control sequence, the method comprising:

displaying a graphical user interface that provides graphical user interface access to a set of motion control operations;

receiving user input to the graphical user interface specifying a sequence of motion control operations;

automatically generating a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, wherein the interconnected nodes visually indicate functionality of the graphical program; and

performing the specified sequence of motion control operations.

2. (Original) The method of claim 1, further comprising:

storing information representing the specified sequence of motion control operations in a data structure in response to said receiving user input specifying the sequence of motion control operations.

3. (Original) The method of claim 2,

wherein the information does not comprise programming language code.

4. (Cancelled)

5. (Original) The method of claim 2,

wherein said receiving user input to the graphical user interface specifying the sequence of motion control operations comprises receiving user input to the graphical

user interface specifying parameter values for one or more motion control operations in the sequence;

wherein said storing information representing the specified sequence of motion control operations comprises storing the parameter values;

wherein said performing the sequence of motion control operations comprises executing software routines corresponding to motion control operations in the sequence, wherein said executing comprises passing the parameter values to the software routines.

6. (Previously presented) The method of claim 1,

wherein said receiving user input to the graphical user interface specifying a sequence of motion control operations does not include receiving user input specifying programming language code to implement the sequence of motion control operations.

7. (Original) The method of claim 1,

wherein the motion control sequence is operable to control motion of a device.

8-10. (Cancelled)

11. (Original) The method of claim 1, further comprising:

receiving user input to the graphical user interface for configuring one or more of the motion control operations in the sequence;

wherein, for each motion control operation, said configuring the motion control operation affects the motion control which the operation is operable to perform.

12. (Original) The method of claim 11,

wherein said receiving user input to the graphical user interface for configuring one or more of the motion control operations in the sequence does not include receiving user input specifying programming language code to configure the motion control operations.

13. (Previously presented) The method of claim 11, further comprising:

for each motion control operation to be configured, displaying a graphical panel including graphical user interface elements for setting one or more properties of the motion control operation and receiving user input to the graphical panel to set one or more properties of the motion control operation.

14. (Original) The method of claim 13, further comprising:

automatically displaying the graphical panel in response to adding the motion control operation to the sequence.

15. (Original) The method of claim 13, further comprising:

receiving user input requesting to configure a first motion control operation; and

displaying a graphical panel for configuring the first motion control operation in response to the request.

16. (Original) The method of claim 1,

wherein the graphical user interface includes an area which visually represents the motion control operations in the sequence;

wherein the method further comprises:

for each motion control operation added to the sequence, updating the area visually representing the motion control operations in the sequence to illustrate the added motion control operation.

17. (Original) The method of claim 16,

wherein the area visually representing the motion control operations in the sequence displays a plurality of icons, wherein each icon visually indicates one of the motion control operations in the sequence;

wherein said updating the area visually representing the motion control operations in the sequence to illustrate the added motion control operation comprises displaying a new icon to visually indicate the added motion control operation.

18. (Original) The method of claim 1,

wherein the graphical user interface displays a plurality of buttons, wherein each button is operable to add a new motion control operation to the sequence in response to user input;

wherein said receiving user input to the graphical user interface specifying the sequence of motion control operations comprises receiving user input to the plurality of buttons to create the sequence of motion control operations.

19. (Original) The method of claim 1,

wherein the set of motion control operations includes:

a straight line move operation;

an arc move operation; and

a contoured move operation.

20. (Original) The method of claim 1, further comprising:

displaying one or more views of the sequence of motion control operations on the graphical user interface, wherein the one or more views graphically preview the cumulative movement specified by the sequence of motion control operations.

21. (Previously presented) The method of claim 20,

wherein the one or more views includes a two-dimensional position view for viewing a two-dimensional display of position data of the sequence in one or more of an XY, YZ, or ZX plane.

22. (Original) The method of claim 20,

wherein the one or more views includes a three-dimensional position view for viewing a three-dimensional display of position data of the sequence.

23. (Previously presented) The method of claim 1, wherein said automatically generating a graphical program implementing the specified sequence of motion control operations comprises programmatically generating a graphical program operable to perform the specified sequence of motion control operations and wherein performing the

specified sequence comprises executing the graphical program to perform the specified sequence of motion control operations.

24. (Cancelled)

25. (Original) The method of claim 23,

wherein the graphical program comprises a graphical data flow program.

26. (Cancelled).

27. (Original) The method of claim 1, further comprising:

receiving a request from a computer program to execute the sequence of motion control operations, wherein the computer program was not used to create the sequence of motion control operations; and

executing the specified sequence of motion control operations in response to the request.

28. (Original) The method of claim 1, further comprising:

programmatically converting the sequence of motion control operations to a format usable for configuring an embedded device to perform the sequence of motion control operations; and

configuring the embedded device to perform the sequence of motion control operations using the format.

29. (Previously Presented) A computer-implemented method for creating a motion control sequence, the method comprising:

displaying a graphical user interface that provides graphical user interface access to a set of motion control operations;

receiving user input to the graphical user interface specifying a sequence of motion control operations;

automatically generating a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, wherein the interconnected nodes visually indicate functionality of the graphical program; and

storing information representing the specified sequence of motion control operations in a data structure.

30. (Original) The method of claim 29,

wherein the information does not comprise programming language code.

31. (Original) The method of claim 29, further comprising:

performing the specified sequence of motion control operations.

32. (Previously Presented) A computer-implemented method for creating a motion control prototype, the method comprising:

receiving user input specifying a desired sequence of motion control operations;

recording the specified sequence of motion control operations in a data structure, wherein the specified sequence of motion control operations comprises the motion control prototype; and

automatically generating a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, wherein the interconnected nodes visually indicate functionality of the graphical program;

wherein the motion control prototype is useable to control a motion device.

33. (Original) The method of claim 32, further comprising:

performing the specified sequence of motion control operations to control the motion device.

34. (Original) The method of claim 32, further comprising:

displaying a set of motion control operations;

wherein the user input comprises user input selecting two or more motion control operations from the set of motion control operations.

35. (Previously Presented) A memory medium for creating a motion control sequence, the memory medium comprising program instructions executable to:

display a graphical user interface that provides graphical user interface access to a set of motion control operations;

receive user input to the graphical user interface specifying a sequence of motion control operations;

automatically generate a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, wherein the interconnected nodes visually indicate functionality of the graphical program; and

perform the specified sequence of motion control operations.

36. (Original) The memory medium of claim 35, further comprising program instructions executable to:

store information representing the specified sequence of motion control operations in a data structure in response to said receiving user input specifying the sequence of motion control operations.

37. (Original) The memory medium of claim 36,

wherein the information does not comprise programming language code.

38. (Original) The memory medium of claim 36, further comprising program instructions executable to:

access the information representing the sequence of motion control operations to determine program instructions corresponding to motion control operations in the sequence; and

execute the program instructions, wherein said performing the specified sequence of motion control operations comprises executing the program instructions.

39. (Original) The memory medium of claim 36,

wherein said receiving user input to the graphical user interface specifying the sequence of motion control operations comprises receiving user input to the graphical user interface specifying parameter values for one or more motion control operations in the sequence;

wherein said storing information representing the specified sequence of motion control operations comprises storing the parameter values;

wherein said performing the sequence of motion control operations comprises executing software routines corresponding to motion control operations in the sequence, wherein said executing comprises passing the parameter values to the software routines.

40. (Original) The memory medium of claim 35,

wherein said receiving user input to the graphical user interface specifying a desired sequence of motion control operations does not include receiving user input specifying programming language code to implement the sequence of motion control operations.

41. (Original) The memory medium of claim 35,

wherein the motion control sequence is operable to control a device to move an object.

42. (Previously Presented) A system for creating a motion control sequence, the system comprising:

a processor;

a memory storing program instructions;

a display device;

wherein the processor is operable to execute the program instructions stored in the memory to:

display a graphical user interface that provides graphical user interface access to a set of motion control operations on the display device;

receive user input to the graphical user interface specifying a sequence of motion control operations;

automatically generate a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes, wherein the interconnected nodes visually indicate functionality of the graphical program; and

execute the specified sequence of motion control operations.

43. (Original) The system of claim 42, further comprising:

a motion control device;

wherein said processor executing the program instructions to execute the specified sequence of motion control operations comprises the processor executing the specified sequence of motion control operations to control the motion control device.

44. (Previously Presented) A system for creating a motion control sequence, the system comprising:

means for displaying a graphical user interface that provides graphical user interface access to a set of motion control operations;

means for receiving user input to the graphical user interface specifying a sequence of motion control operations;

means for automatically generating a graphical program implementing the specified sequence of motion control operations, wherein automatically generating the graphical program comprises automatically including a plurality of nodes in the graphical program and automatically generating a plurality of connections between the nodes,

wherein the interconnected nodes visually indicate functionality of the graphical program; and

means for performing the specified sequence of motion control operations.

45. (Previously presented) The method of claim 1, further comprising compiling the graphical program into executable compiled code.

46. (Previously presented) The method of claim 1, wherein receiving user input specifying a sequence of motion control operations comprises the user selecting at least two motion control icons, and wherein the automatically generated graphical program is distinct from the motion control icons.

47. (Previously presented) The method of claim 1, wherein the automatically generated graphical program is modifiable by a user without the user having to modify the sequence of motion control operations.

48. (Previously Presented) The method of claim 1, wherein automatically generating the graphical program includes automatically generating the plurality of nodes and the plurality of connections between the nodes without direct user input specifying nodes or connections between the nodes.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.